

In the Claims:

Please cancel claims 1-8 and 10-21; and add new Claims 22-41, all as shown below.

1 - 21. (Cancelled).

22. (New) A system to perform distributed transaction processing, comprising:

a plurality of resources that are associated with one or more transactions;

a transaction manager on one or more server, wherein the one or more server manages a thread pool with a plurality of server threads, and wherein the transaction manager uses the thread pool to perform transaction operations on the plurality of resources, wherein the transaction operations are in different groups and each transaction operation in each group targets to a transaction operation on one of the plurality of resources,

wherein, for each group of the transaction operations on the plurality of resources, the transaction manager dispatches all but one transaction operations in the group to available server threads except a primary thread, wherein the primary thread process the remaining one transaction operation and receives signals from the other server threads that indicate the completion of dispatched transaction operations before the transaction manager dispatches a next group of transaction operations.

23. (New) The system of Claim 22, wherein:

the primary thread process commands relating to dispatch requests that were unable to be dispatched to separate threads.

24. (New) The system of Claim 22, wherein:

one of the groups of transaction operations is associated with a prepare phase of a transaction.

25. (New) The system of Claim 22, wherein:

one of the groups of transaction operations is associated with a commit phase of a transaction.

26. (New) The system of Claim 22, wherein:

each one of the transaction operations in one group is performed in parallel.

27. (New) The system of Claim 22, wherein:
if an idle thread does not exist, the thread pool manager processes the transaction operations in one group in sequence.
28. (New) The system of Claim 22, wherein:
the primary thread wait for a pre-determined period of time during which the threads receiving dispatches signal the primary thread.
29. (New) The system of Claim 22, wherein:
the signals are handled based on a class that maintains a counter of signal events.
30. (New) The system of Claim 22, wherein:
each request that is dispatched to a thread for execution is given a reference to a synchronization object.
31. (New) The system of Claim 22, wherein:
the transaction operations are dispatched in three or more groups.
32. (New) A method to perform distributed transaction processing, comprising:
associating a plurality of resources with one or more transactions;
providing a transaction manager on one or more server, wherein the one or more server manages a thread pool with a plurality of server threads;
organizing transaction operations on the plurality of resources in different groups, wherein each transaction operation in each group targets to a transaction operation on one of the plurality of resources,
performing one group of transaction operations on the plurality of resources, wherein the transaction manager dispatches all but one transaction operations in the group to available server threads except a primary thread, wherein the primary thread process the remaining one transaction operation and wait for signals from the other server threads that indicate the completion of dispatched transaction operations; and
after the primary thread receives signals from the other server threads that indicate the completion of dispatched transaction operations, dispatching a next group of transaction operations.
33. (New) The method of Claim 32, wherein:

the primary thread process commands relating to dispatch requests that were unable to be dispatched to separate threads.

34. (New) The method of Claim 32, wherein:
one of the groups of transaction operations is associated with a prepare phase of a transaction.
35. (New) The method of Claim 32, wherein:
one of the groups of transaction operations is associated with a commit phase of a transaction.
36. (New) The method of Claim 32, wherein:
each one of the transaction operations in one group is performed in parallel.
37. (New) The method of Claim 32, wherein:
if an idle thread does not exist, the thread pool manager processes the transaction operations in one group in sequence.
38. (New) The method of Claim 32, wherein:
the primary thread wait for a pre-determined period of time during which the threads receiving dispatches signal the primary thread.
39. (New) The method of Claim 32, wherein:
the signals are handled based on a class that maintains a counter of signal events.
40. (New) The method of Claim 32, wherein:
each request that is dispatched to a thread for execution is given a reference to a synchronization object.
41. (New) The method of Claim 32, wherein:
the transaction operations are dispatched in three or more groups.